SPARSE DENOISING OF AUDIO BY GREEDY TIME-FREQUENCY SHRINKAGE

Gautam Bhattacharya, Philippe Depalle

McGill University, Schulich School of Music & CIRMMT
555 Sherbrooke St. Ouest, Montreal, Quebec, Canada, H3A 1E3

ABSTRACT

Matching Pursuit (MP) is a greedy algorithm that iteratively builds a sparse signal representation. This work presents an analysis of MP in the context of audio denoising. By interpreting the algorithm as a *simple shrinkage* approach, we identify the factors critical to its success, and propose several approaches to improve its performance and robustness. We present experimental results on a wide range of audio signals, and show that the method is able to yield results thats are competitive with other audio denosing approaches. Notably, the proposed approach retains a small percentage of the transform signal coefficients in building a denoised representation, i.e., it produces very *sparse* denoised results.

Index Terms— Matching Pursuit, Greedy Search, Simple Shrinkage, Sparse Representation, Audio Denoising

1. INTRODUCTION

Time-Frequency audio denoising approaches remove unwanted noise from audio signals by attenuating it in the frequency domain. The idea is that for audio like speech and music, most of the meaningful parts of the sound are concentrated in a relatively small number of frequency components. As a result, one can attenuate or *shrink* the noise in the frequency domain, while leaving the meaningful parts of the sound largely unaffected [1]. Modern signal denoising approaches are often equipped with an assumption of *sparsity* [2, 3, 4, 5], which refers to the circumstance that many natural signals can be expanded (using a suitable dictionary Φ) with only few non-zero coefficients. Matching Pursuit (MP) is one such algorithm, that iteratively selects time-frequency atoms from a dictionary Φ, to build a sparse signal representation [6]. Despite its ability to yield sparse representations, MP has largely been unexplored in the context of audio/signal denoising. One of the reasons for this, perhaps, is that sparsity is not enforced in MP, but rather a consequence of a *greedy search* – which iteratively selects the most energetic dictionary atom, i.e., the atom most correlated with the residual signal. This paper presents an analysis of MP denoising in the context of audio noise reduction, as well as several strategies to improve its performance and robustness. In doing so we introduce a new audio denoising approach called *Greedy Time-Frequency Shrinkage* (GTFS), that we will show is able to produce competitive denoising results in terms of standard performance metrics, SNR and PEAQ[1] [7]. While many forms of noise exist, we focus on the removal of uncorrelated Gaussian white noise from music and speech signals.

[1]SNR - Signal to Noise Ratio, PEAQ - Perceptual Evaluation of Audio Quality

2. GREEDY TIME-FREQUENCY SHRINKAGE

The denoising principle in MP is based on the fact that the algorithm selects time-frequency atoms that are highly correlated with the signal in order to build a signal representation. Thus as the noise we wish to remove is uncorrelated, MP will first select the correlated deterministic atoms before it selects the noisy ones. The success of the approach is based on MP being able to identify when a noisy atom is selected, and stopping the decomposition process. A natural first approach to achieve this is by applying a threshold on the correlation value between the dictionary atoms and the residual MP signal. When MP is used to denoise signals in this way, it can be interpreted as a *simple shrinkage* approach. We hence dub this denoising method – Greedy Time-Frequency Shrinkage (GTFS). Wavelet shrinkage, or just, Shrinkage, is a classical denoising method in signal processing. It is based on the idea that an oracle furnishes information about how best to adapt a spatially variable estimator, to an unknown function [8, 9, 10].

2.1. Signal Model

We consider an observation signal $f \in \mathcal{R}^N$ that is corrupted by Gaussian white noise $e \sim \mathcal{N}(0, \sigma^2)$:

$$y[n] = f[n] + e[n] \qquad n = 0, ..., N-1$$

Where it is assumed that $f = \Phi c$, with sparse synthesis coefficients c and redundant time-frequency dictionary Φ.

2.2. GTFS Algorithm

The MP residual signal is initialized as $r_0 = y$. At each iteration MP selects the time-frequency atom with the highest correlation to the residual signal and assigns it a weight, $\alpha_i = \langle g_{m(i)}, r_i \rangle$, where $g_{m(i)}$ is the selected dictionary atom and r_i is the residual signal at the i^{th} iteration. This coefficient α_i is then shrunk or attenuated based on a hard thresholding rule [8], with threshold λ :

$$\alpha_H(\alpha_i, \lambda) = \alpha_i \{|\alpha_i| > \lambda\} \tag{1}$$

If the threshold condition is met, the weighted atom is subtracted from the residual signal and the process is repeated until a stopping condition is met. After N iterations the signal is recovered as $\hat{f} = \sum_{i=0}^{N-1} \alpha_i g_{m(i)}$.

Ideally, the algorithm should stop when all the deterministic audio components have been recovered. However this is not always the case, as GTFS denoising is sensitive to the shrinkage operator, the threshold value λ, and the dictionary used for signal analysis.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

2.3. Shrinkage Operator

GTFS or MP denoising based on correlation thresholding, iteratively applies the hard thresholding rule to each greedily selected atom. The hard thresholding operator is a discontinuous function, and produces results with high variance [11]. Moreover, hard thresholding is a *diagonal estimation* approach, wherein each time-frequency coefficient is processed independently. One of the known drawbacks of diagonal estimators in audio denoising is that they produce isolated time-frequency artifacts that are perceived as *musical noise* [12]. The hard thresholding operator does not actually attenuate or shrink atomic coefficients, i.e., it is a 'keep' or 'kill' rule, and hence the success of the approach depends heavily on the threshold value used.

2.4. Threshold Selection

In order to perform shrinkage on the atomic coefficients of a noisy signal, an appropriate threshold value is needed. Donoho and Johnstone introduced the *Universal, Minimax and SURE thresholds* for thresholding the transform coefficients of a noisy signal [8, 13]. The table below compares the denoising performance of each of these thresholds in conjunction with GTFS. The audio used for this experiment was a solo trumpet recording sampled at 44.1 kHz, deteriorated with 5dB white noise.

Threshold	SNR(dB)
Universal	16.16
Minimax	**18.61**
SURE	17.70

Table 1. Denoising results for different thresholds.

The Universal threshold is considerably larger than the Minimax and SURE thresholds, and consequently stops the signal decomposition process the earliest, producing the sparsest result. While it is able to completely attenuate the noise, the Universal threshold underfits the data, i.e., it is unable to *completely* recover the sound. As a result it yields the lowest output SNR of the three thresholds. That being said, both the Minimax and SURE thresholds used in conjunction with GTFS produce high-frequency musical noise artifacts.

2.5. Dictionary Design

Signal transformation in MP is achieved by decomposing a signal over a redundant dictionary. The main assumption is that the signal under analysis can be represented by a small number of dictionary elements, i.e., an assumption of *sparsity*. In the context of audio denoising, in addition to sparsely representing the sound, the dictionary should not be correlated to the noisy signal content. Thus, GTFS denoising is highly dependent on the dictionary, and the diagonal estimation approach often leads to data overfitting and musical noise artifacts. Numerical experiments have shown that GTFS produces the best denoising results when a dictionary with relatively long atoms is used. This is because long (tonal) atoms are well correlated to the tonal content of a sound. Longer atoms are also less likely to match noisy signal content than shorter atoms. Another advantage of using longer atoms is that it implies sparser denoising results. In general, sparse denoised results imply good results in terms of musical noise attenuation and output SNR. However in extreme cases, underfitting leads to results that are *too* sparse, and generally unacceptable. We make use of a three scale dictionary of Gabor atoms corresponding to window lengths of 8192, 4096 and 2048 samples for the majority of our experiments. It should be noted that while this kind of dictionary worked well for the majority of our experiments, it underfits the data for sounds with a large amount of transient content, i.e., sharp attacks, frequency modulation, breath noise etc. For such sounds we repeatedly perform a decomposition on the residual signal using single-scale dictionaries with short atoms, corresponding to window lengths of either 512, 256 or 128 samples.

2.6. Stopping Criteria

A simple criterion for stopping a MP decomposition is based on the ratio of the energies of the reconstructed and residual signals, known as the signal-to-residual ratio (SRR). Once the SRR falls beneath a threshold, typically a very small value, the process is stopped. In GTFS denoising, this typically occurs when an atom does not meet the correlation threshold condition.The idea is that the algorithm should stop once all the meaningful or non-noisy audio content has been recovered.

3. MODEL ENHANCEMENTS

Having identified the primary factors on which GTFS denoising depends, we present several approaches to improve its performance in terms of musical noise attenuation, output SNR and algorithmic stability. The dictionary used for signal transformation plays a central role in GTFS denoising, and can suffer from problems of both over and underfitting. The underfitting problem can be largely avoided by using a very redundant dictionary. This however often leads to data overfitting and musical noise artifacts. In order to tackle this problem we consider different rules to attenuate the transform coefficients of the noisy signal, and introduce an iterative thresholding strategy based on the SURE threshold.

3.1. Attenuation Rule

GTFS makes use of the hard thresholding rule, that does not attenuate a coefficient if it satisfies the threshold. Thus if an erroneous atom gets past the correlation threshold, there is no provision to reduce or nullify its influence. In order to tackle this problem, we consider two alternate attenuation rules.

3.1.1. Soft Thresholding

Along with hard thresholding, Donoho and Johnstone introduced a soft thresholding operator for the shrinkage of wavelet coefficients [14]. Unlike hard thresholding, the soft thresholding operator is a continuous function, and produces results with high bias due to attenuation of large coefficients [11]. The GTFS-ST algorithm maintains the same GTFS denoising structure, replacing the hard thresholding step with a soft threshold :

$$\alpha_S(\alpha_i, \lambda) = sgn(\alpha_i)(|\alpha_i| - \lambda)_+ \quad (2)$$

Where $(.)_+$ represents $max(0, .)$. The GTFS-ST algorithm was tested with the same dictionary and trumpet recording as before. We make use of the SURE and Minimax thresholds, as GTFS using these thresholds produced musical noise artifacts. GTFS-ST yielded a denoised signal with an SNR of 16.81dB and 17.15dB respectively. The algorithm was able to attenuate musical noise to a *much* greater extent than GTFS, with complete removal of musical noise in the Minimax case. The GTFS-ST was able to greatly attenuate musical noise with the SURE threshold as well, however a small amount of audible artifacts remained in the denoised sound.

3.1.2. *Empirical Wiener Attenuation*

Like simple shrinkage, Empirical Wiener filtering is a signal denoising approach based on diagonal estimation [15], and has been shown to produce results that are somewhat in between the hard and soft thresholding operators [16]. The Empirical Wiener attenuation rule is given by

$$\alpha_{EW}(\alpha_i, \lambda) = \alpha_i \left(1 - \left[\frac{\lambda}{|\alpha_i|}\right]^2\right)_+ \quad (3)$$

We tested the empirical Wiener attenuation rule under the same conditions as the soft thresholding operator. The Wiener rule was able to greatly attenuate musical noise with both the Minimax and SURE thresholds. Interestingly, GTFS with the empirical Wiener rule underfit the data when used in conjunction with the Minimax threshold, achieving a SNR of 14.52dB. With the SURE threshold GTFS-EW yielded a denoised result with a SNR of 17.26dB.

3.2. Dynamic Thresholding

In the GTFS denoising framework, the wavelet thresholds are *static*, i.e., they remain the same throughout the signal denoising process. This is illustrated by the three straight lines in Fig. 1. While these thresholds worked quite well with the soft thresholding and empirical Wiener rules, they yield musical noise artifacts when used in conjunction with hard thresholding. In this section we develop dynamic thresholding strategies in which thresholds are iteratively updated based on the changing residual signal.

3.2.1. *GreedySURE*

In the GreedySURE approach, for each iteration of the GTFS algorithm, the SURE threshold is calculated based on the transform coefficients of the residual signal. The dashed line (red) in fig. 1 shows the evolution of the GreedySURE threshold over the course of a GTFS denoising decomposition.

3.2.2. *NaiveSURE*

The NaiveSURE approach is similar to GreedySURE, with the difference that the SURE threshold is computed on the noisy residual signal itself, and not on its transform coefficients. The green curve in fig. 1 represents the NaiveSURE threshold, which starts out noticeably larger than all the other thresholds.

3.2.3. *BlockSURE*

BlockSURE is a variation of GreedySURE that determines thresholds based on local information. Rather than compute the SURE threshold based on the transform coefficients of the *entire* residual signal at each iteration, we construct a time-frequency block around the selected atom and evaluate the SURE threshold based on that block. This approach is similar to the adaptive thresholding approach used in audio block thresholding [17]. Numerical experiments have shown that best results are achieved with blocks corresponding to 16 units in time and 8 units in frequency. For atoms of length 2048 samples, this corresponds to 743 ms time blocks, and a frequency range of 172.26 Hz. The BlockSURE threshold is represented by the black curve in fig. 1, and overlaps with the GreedySURE curve (red) most of the time.

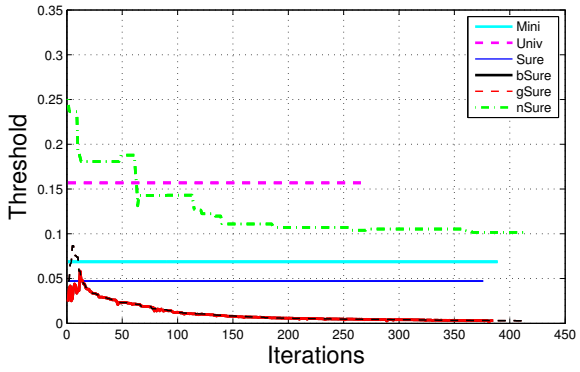

Fig. 1. Evolution of Static and Dynamic Thresholds over the course of a GTFS denoising process.

Thr	Mini	Univ	Sure	gSure	nSure	bSure
SNR	18.54	16.55	17.79	18.40	18.67	**19.21**

Table 2. Denoising results for noisy flute recording (5dB noise) with various thresholds. Mini = Minimax, Uni = Universal, Sure = SURE, gSure = GreedySURE, nSure = NaiveSURE, bSure = BlockSURE.

3.2.4. *Discussion*

Both the soft thresholding operator and the empirical Wiener attenuation rule are able to attenuate musical noise to a larger extent than hard thresholding. Apart from affecting the contribution of each atom, the choice of attenuation rule also affects the GTFS denoising process in terms of frequency of the atoms selected. Interestingly, the three attenuation rules select mostly the same set of frequencies – roughly 90%. This implies that a very small fraction of time-frequency coefficients influence whether the algorithm will overfit the data or not.

While the soft thresholding and empirical Wiener rules help with the overfitting problem, both methods occasionally yield relatively low SNR. In order to improve performance in this regard, we developed dynamic thresholding approaches. Fig. 1 shows the evolution of the various dynamic thresholds. Each of them displays a decreasing trend, which helps to avoid the underfitting problem in general. The dynamic thresholds also perform better than their static counterparts in terms of attenuating musical noise. Table 2 compares the denoising results for the various static and dynamic thresholding strategies in terms of SNR. The audio used for this test was a solo flute recording deteriorated by 5dB noise. Each threshold was tested in conjunction with the hard thresholding rule, as it displayed the greatest tendency to overfit data. We confirm that the static Minimax and SURE thresholds, as well as the the dynamic GreedySURE approach all produced musical noise. However the NaiveSURE and BlockSURE approaches were able to greatly attenuate musical noise artifacts, while achieving high SNR (i.e. not underfitting the data).

4. EXPERIMENTS AND RESULTS

In the previous section we presented a variety of approaches to improve the performance of MP based audio denoising in terms of attenuating musical noise, and achieving higher output SNR. The

SNR	Violin	Flute	Piano	M-Voice	F-Voice
BT	**14.87**	**20.31**	**18.60**	**12.94**	**16.95**
PEW	14.10	18.67	16.54	11.26	15.28
GTFS	14.18	19.21	16.74	12.00	15.60

PEAQ	Violin	Flute	Piano	M-Voice	F-Voice
BT	-3.63	-3.06	-3.04	-2.92	-3.30
PEW	**-2.83**	**-2.39**	**-2.20**	**-2.39**	-2.36
GTFS	-3.01	-2.62	-2.42	-2.55	**-2.29**

Table 3. SNR & PEAQ (5dB noise). BT = Block Thresholding, PEW = Persistent Empirical Wiener, GTFS = Greedy Time Frequency Shrinkage.

SNR	Violin	Flute	Piano	M-Voice	F-Voice
BT	**12.34**	**17.09**	**15.18**	**10.30**	**13.74**
PEW	10.04	16.20	12.84	08.50	11.89
GTFS	09.73	16.47	13.72	09.26	12.20

PEAQ	Violin	Flute	Piano	M-Voice	F-Voice
BT	-3.40	-3.36	-3.06	-2.86	-3.31
PEW	-2.68	**-2.64**	**-1.95**	-2.36	-2.42
GTFS	**-1.78**	-2.83	-2.58	**-2.00**	**-2.17**

Table 4. SNR & PEAQ (0dB noise). BT = Block Thresholding, PEW = Persistent Empirical Wiener, GTFS = Greedy Time Frequency Shrinkage.

unpredictable nature of the greedy atomic decomposition process, makes experimental testing and analysis an important aspect of developing and improving the approach. In this section we test the GTFS denoising approach on a wide range of audio signals. Algorithmic performance is measured in terms of output SNR and PEAQ scores. The PEAQ metric scores the perceptual distortion in the denoised audio on a scale ranging from a minimum of -4 (worst case, i.e. maximum perceptible distortion) up to a maximum of 0 (best case, i.e. no perceptible distortion at all).

4.1. Comparison with other Algorithms

In this case study we compare the performance of the GTFS denoising approach with two other algorithms - audio Block Thresholding (BT) [17] and Persistent Empirical Wiener (PEW) denoising [18, 19]. We compare the algorithms over a range of audio signals which include solo recordings of the violin, flute and piano, as well as male (M-Voice) and female (F-Voice) speech signals. All audio was sampled at 44.1 kHz. Testing is done at noise levels of 5dB and 0dB respectively.

Algorithm	5dB Noise	0dB Noise
BT	97.62	94.45
PEW	14.33	11.04
GTFS	**00.19**	**00.07**

Table 5. Average Number of Transform Coefficients Retained (%).

From Table 3 & 4 we see that GTFS denoising produces competitive results with both BT and PEW in terms of SNR and PEAQ scores, at different noise levels. While there is much debate regarding the merits of different performance metrics, we can confirm that the denoised results produced by all three algorithms are of comparable quality [2]. This assessment is based on informal listening tests. Notably (cf. Table 5), GTFS retains a significantly smaller number of the transform signal coefficients than both BT and PEW, retaining between 0.07-0.19% of the transform coefficients. In our testing the PEW denoising approach retained 11.04-14.33% of the coefficients on average. BT is not a sparse denoising approach and retains most of the transform coefficients.

4.2. Computational Complexity

The GTFS algorithms that make use of a static thresholding strategy. take approximately the same amount of time to perform a (denoised) signal decomposition than a standard MP would do. The only additional step involves the calculation of the threshold value. The dynamic thresholding approaches on the other hand, take slightly longer to process, as a threshold value has to be calculated at each iteration. We note in our experiments, however, that GTFS generally produces a much sparser signal representation, which reduced number of iterations compensates for the per-iteration threshold computation. A comparison of the computation times for the different GTFS approaches is provided on the companion website.

5. CONCLUSION

This work presents an analysis of Matching Pursuit based signal denoising by casting the problem as a *Greedy Time-Frequency Shrinkage*. We identified dictionary design and threshold selection as key components in the success of the approach, and proposed using alternate attenuation rules and dynamic thresholding strategies in order to enhance its output SNR, attenuation of musical noise and algorithmic stability. GTFS denoising was shown yield results that are comparable to other state-of-the-art audio denoising algorithms in terms of PEAQ and SNR scores. One of the advantages of the GTFS framework is that it is easily extendible. Future work will attempt to extend GTFS denoising framework to non-stationary noise, incorporate non-diagonal estimators and different attenuation rules [20, 21, 22, 23, 4].

6. REFERENCES

[1] Simon Godsill, Peter Rayner, and Olivier Cappé, "Digital audio restoration," in *Applications of Digital Signal Processing to Audio and Acoustics*, Mark Kahrs and Karlheinz Brandenburg, Eds. Springer, 1998.

[2] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[3] Robert Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[4] Tony Cai, "Adaptive wavelet estimation: a block thresholding and oracle inequality approach," *The Annals of Statistics*, vol. 27, no. 3, pp. 898–924, 1999.

[2] denoised examples of all three approaches are available at : http://music.mcgill.ca/ gautam/ICASSP/icassp2.html

[5] Michael Elad, "Why simple shrinkage is still relevant for redundant representations?," *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5559–5569, 2006.

[6] Stéphane Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[7] Peter Kabal, "An examination and interpretation of ITU-R BS. 1387: Perceptual evaluation of audio quality," *TSP Lab Technical Report, Dept. Electrical & Computer Engineering, McGill University*, pp. 1–89, 2002.

[8] David L. Donoho and Jain M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[9] David L. Donoho, Iain M. Johnstone, Gérard Kerkyacharian, and Dominique Picard, "Wavelet shrinkage: asymptopia?," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 301–369, 1995.

[10] David L. Donoho and Iain M. Johnstone, "Minimax estimation via wavelet shrinkage," *The Annals of Statistics*, vol. 26, no. 3, pp. 879–921, 1998.

[11] Hong-Ye Gao, "Wavelet shrinkage denoising using the non-negative garrote," *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 469–488, 1998.

[12] Olivier Cappé, "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 345–349, 1994.

[13] David L. Donoho and Iain M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[14] David L. Donoho, "De-noising by soft-thresholding," *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 613–627, 1995.

[15] Robert McAulay and Marilyn Malpass, "Speech enhancement using a soft-decision noise suppression filter," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 2, pp. 137–145, 1980.

[16] Kai Siedenburg and Monika Dörfler, "Audio denoising by generalized time-frequency thresholding," in *Audio Engineering Society Conference: 45th International Conference: Applications of Time-Frequency Processing in Audio*. Audio Engineering Society, 2012.

[17] Guoshen Yu, Stéphane Mallat, and Emmanuel Bacry, "Audio denoising by time-frequency block thresholding," *Signal Processing, IEEE Transactions on*, vol. 56, no. 5, pp. 1830–1839, 2008.

[18] Kai Siedenburg, "Persistent empirical Wiener estimation with adaptive threshold selection for audio denoising," in *Proceedings of the 9th Sound and Music Computing Conference, Copenhagen (July 11-14th)*, 2012.

[19] Kai Siedenburg and Monika Dörfler, "Persistent time-frequency shrinkage for audio denoising," *Journal of the Audio Engineering Society*, vol. 61, no. 1/2, pp. 29–38, 2013.

[20] Israel Cohen and Baruch Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, vol. 81, no. 11, pp. 2403–2418, 2001.

[21] Israel Cohen, "Speech enhancement using a noncausal a priori SNR estimator," *Signal Processing Letters, IEEE*, vol. 11, no. 9, pp. 725–728, 2004.

[22] Yariv Ephraim and David Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 32, no. 6, pp. 1109–1121, 1984.

[23] Yariv Ephraim and David Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 33, no. 2, pp. 443–445, 1985.